# Neuro-fuzzy Speed Controller for DC Drives Using Low Precision Shaft Encoder

**Ferenc Farkas, Sándor Halász, István Kádár**

Department of Electric Power Engineering
Budapest University of Technology and Economics
Egry József út 18, H-1111 Budapest, Hungary, ikadar@eik.bme.hu

*Abstract: The speed is generally computed from the pulse number of a shaft encoder obtained in a sampling period. However, the speed resolution is not suitable for speed control feedback in the lower speed range, when a very short sampling period is used. On the other hand, the shorter the sampling time the better the control quality is. To overcome this contradiction a neuro-fuzzy speed controller is proposed which has as a feedback not only the encoder information, but the measured armature current and armature voltage. Thus, the load, and the speed estimation is more accurate than in PID speed controllers.*

*Keywords: DC servomotor, neuro-fuzzy speed controller, load estimation*

## 1 Introduction

In most servo systems the speed information is usually calculated from the impulse number of a shaft encoder between successive samples. More precisely, at $[m]$th sampling point the speed is calculated using relation

$$\omega[m] = \frac{\Theta[m] - \Theta[m-1]}{T_1} = \frac{2\pi}{N_0} \cdot \frac{N[m] - N[m-1]}{T_1}, \tag{1}$$

where $\Theta[m]$, $\Theta[m-1]$, and $N[m], N[m-1]$ represent the positions and pulse numbers at $[m]$th, $[m-1]$th sampling points, respectively, while $N_0$ represents the precision of the shaft encoder (number of impulses per revolution), and $T_1$ is the sampling time of the controller.

The speed resolution is the lowest speed change which can be observed when at least one pulse is obtained between successive sampling points:

$$\Delta\omega = \frac{2\pi}{N_0 \cdot T_1}. \tag{2}$$

Thus, the speed resolution is inverse proportional to the sampling period $T_1$ and to the precision of the shaft encoder $N_0$. In Table 1 some speed resolution values are given for different sampling periods and encoder precisions.

Table 1
Speed resolution $\Delta\omega(T_1, N_0)$

| Speed resolution $\Delta\omega$ [rpm] | Sampling period $T_1$ [ms] | Encoder precision $N_0$ [impulse/rev] |
|---|---|---|
| 0.6 | 10 | 10000 |
| 6 | 1 | 10000 |
| 60 | 0.1 | 10000 |
| 2.4 | 10 | 2500 |
| 24 | 1 | 2500 |
| 240 | 0.1 | 2500 |

As it can be observed the speed resolution is not suitable for speed control feedback, particularly in the lower speed range, when a very short sampling period is used. In order to overcome this problem, a longer sampling period may be used, but in that case the system may become unstable due to a longer dead-time inserted into the control loop.

To overcome these problems an instantaneous speed observer is proposed in [2], [3] by Hori. However, experiences show that the change of load is estimated with a relatively good precision only after (3-4) $T_1$. In this article, a neuro-fuzzy speed controller is proposed which uses not only the encoder signal and the armature current, but the armature voltage as well. Thus, the proposed controller estimates the change of load better than the speed observer proposed in [2], [3].

## 2 Theoretical Considerations

In Figure 1 the timing chart of the system is shown. The counter of the shaft

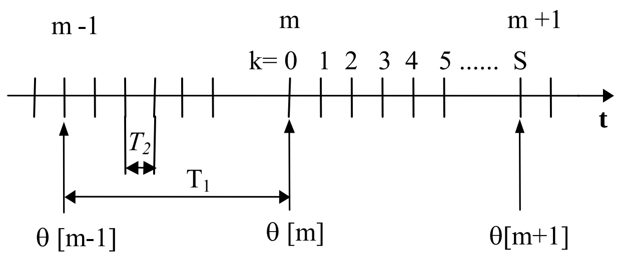



Figure 1
Timing chart

encoder is read out every $T_1$ period. That is, at every $T_1$ the shaft position $\Theta[m]$ of the DC motor is measured. Time $T_1$ is several times longer than the control period $T_2$ of the neuro-fuzzy controller to have a suitable speed resolution. At the shorter sampling points of $T_2$ (represented by $k$=0, 1, 2, ...,S) the total acceleration torque $\hat{M}[m,k]$ is given by the sum of motor torque and the estimated disturbance torque:

$$\hat{M}[m,k] = K_n \cdot i[m,k] + \hat{M}_d[m], \tag{3}$$

where, the ^ symbol indicates an estimated value, $\hat{M}_d[m]$ represents the estimated load considered constant during $T_2$ period, while $i[m,k]$ and $K_n$ represent the measured armature current and the torque coefficient of the DC motor, respectively. The estimated speed will be given by the following formula considering a linear approximation for the acceleration torque:

$$\hat{\omega}[m,k] = \hat{\omega}[m,k-1] + \frac{T_2}{J_n} \cdot \frac{\hat{M}[m,k] + \hat{M}[m,k-1]}{2}, \tag{4}$$

where $J_n$ represents the moment of inertia. The linear approximation is a good approach when $T_1$ is in the order of 10 ms. Writing the equation (4) for different $k$=1, 2, …, S values, and summing these equations the estimated speed at sampling point $k$=S is obtained in function of acceleration torque and the initial speed (speed at $k$=0).

$$\hat{\omega}[m,S] = \hat{\omega}[m,0] + \frac{T_2}{J_n} \cdot \sum_{k=1}^{S} \frac{\hat{M}[m,k] + \hat{M}[m,k-1]}{2}. \tag{5}$$

Substituting equation (3) into (5) the following relation is obtained:

$$\hat{\omega}[m,S] = \hat{\omega}[m,0] + T_1 \cdot \frac{K_n I[m] + \hat{M}_d[m]}{J_n}, \tag{6}$$

where the following notation is applied:

$$I[m] = \frac{1}{2S} \cdot \sum_{k=1}^{S} (i[m,k] + i[m,k-1]). \tag{7}$$

It should be noted that $I[m]$ represents the average armature current which accelerates the motor during $T_1$. This is easy to see if $i[m,k]$ is set to a constant value in equation (7), that is $i[m,k] = I[m]$.

The estimated position is calculated from the estimated speed using linear approximation:

$$\hat{\Theta}[m,k] = \hat{\Theta}[m,k-1] + \frac{T_2}{2} \cdot \left(\hat{\omega}[m,k] + \hat{\omega}[m,k-1]\right). \tag{8}$$

Writing the equation (8) for different $k$ =1, 2, …, S values and summing these equations the estimated position at sampling point $k$ =S is obtained in function of estimated speed and initial position (position at $k$ =0).

$$\hat{\Theta}[m,S] = \hat{\Theta}[m,0] + T_1 \cdot \hat{\Omega}[m], \tag{9}$$

where the following notation is applied:

$$\hat{\Omega}[m] = \frac{1}{2S} \cdot \sum_{k=1}^{S} (\hat{\omega}[m,k] + \hat{\omega}[m,k-1]) . \tag{10}$$

It should be noted that $\hat{\Omega}[m]$ represents the average rotor speed producing the position change during $T_1$. This is easy to see if $\hat{\omega}[m,k]$ is set to a constant value in equation (10), that is $\hat{\omega}[m,k] = \hat{\Omega}[m]$.

At $k$ =S there might be a difference between the estimated position and the measured position of the shaft:

$$\Delta\Theta = \hat{\Theta}[m,S] - \Theta[m+1] . \tag{11}$$

Correction on the estimated speed and load can be made at every $T_1$ period using the difference between the estimated and measured position of the rotor. The question is how to modify these estimated values? To answer this question a further investigation has to be done to see why this difference may appear.

The real position of the rotor can be expressed as

$$\Theta[m+1] = \Theta[m] + \Omega[m] \cdot T_1 , \tag{12}$$

where $\Omega[m]$ represents the average constant speed producing the change of rotor position.

Substituting equation (9) and (12) into equation (11) the difference in position is obtained:

$$\Delta\Theta = \hat{\Theta}[m,0] + T_1 \cdot \hat{\Omega}[m] - \Theta[m] - T_1 \cdot \Omega[m] . \tag{13}$$

The speed variation is small when the mechanical time-constant of the servo drive is longer than the $T_1$ sampling time. Thus, the estimated constant speed can be expressed as a linear approximation of the estimated initial and final speed:

$$\hat{\Omega}[m] = \frac{\hat{\omega}[m,0] + \hat{\omega}[m,S]}{2} . \tag{14}$$

Substituting equation (9) into (14) it is obtained:

$$\hat{\Omega}[m] = \hat{\omega}[m,0] + \frac{T_1}{2} \cdot \frac{K_n I[m] + \hat{M}_d[m]}{J_n} . \tag{15}$$

Thus, the position difference is obtained as

$$\Delta\Theta = T_1(\hat{\omega}[m,0] - \Omega[m]) + \frac{T_1^2}{2}\left(\frac{K_n I[m] + \hat{M}_d[m]}{J_n}\right), \tag{16}$$

where the relation $\hat{\Theta}[m,0] = \Theta[m]$ is used.

From equation (16) it can be concluded that the difference between the estimated and measured rotor position is due to the estimation error in the initial speed, load torque, and moment of inertia, respectively. Substituting in equation (16) the terms in brackets with $\Delta\omega$ and $\Delta M_d$, respectively, it can be obtained:

$$\Delta\Theta = T_1\Delta\omega + \frac{T_1^2}{2J_n}\Delta M_d . \tag{17}$$

Thus, a part of the position difference is due to difference in estimated and real average rotor speed, and a part of the position difference is due to the error in load and moment of inertia estimation:

$$\alpha_1 \cdot \Delta\Theta = \frac{T_1}{2J_n}\Delta M_d , \tag{18}$$

$$\alpha_2 \cdot \Delta\Theta = T_1\Delta\omega . \tag{19}$$

The main disadvantage of the proposed algorithm is that in case of abrupt change of load, the controller needs at least (3-4) $T_1$ time to estimate the changed load with a good approximation. This is due to the fact that only one known value – difference in rotor position – is used to made corrections to two different estimated values. Thus, a second relation is needed to have a better estimation to both speed and load. The following relation is proposed for the second equation:

$$K_n\hat{\omega}[m,k] = u[m,k] - R_a i[m,k] - L_a \frac{i[m,k] - i[m,k-1]}{T_2} , \tag{20}$$

where $R_a$, $L_a$ are the armature resistance, and inductance respectively, while $u[m,k]$ is the armature voltage. However, measuring $u[m,k]$ is not an easy task when 4 quadrant chopper amplifier is used. In the next section a neural network is shown, which is trained to model equation (20).

# 3 Estimating Speed with Neural Network

In case of 4 quadrant chopper the armature voltage varies between $-U_n$ and $+U_n$, or between $-U_n$, $0$, and $+U_n$. Thus, measuring $u[m,k]$ is a challenging task. Using a 20 kHz chopper, the armature voltage might change at every 50 µs. However, $T_2$ is in order of 100 µs, which means the instantaneous voltage should be measured instead of the average one. Using an RC integration circuit the average armature voltage can be obtained by hardware. An important question is what should be the value of R and C, or more importantly the time-constant of the RC circuit?

As it can be seen in Figure 2 the armature voltage varies between $-U_n$ and $+U_n$. In order to have a good estimation of the armature voltage, a compromise between measuring the armature voltage in the steady-state and or in transient state must be made. In steady-state (the upper diagram in Figure 2) the armature voltage variation can be seen as a noise added to the direct voltage. Thus, we would like to filter out this variation as much as possible. However, in transient state (the lower diagram in Figure 2) the measured voltage should follow the armature voltage as quick as possible.

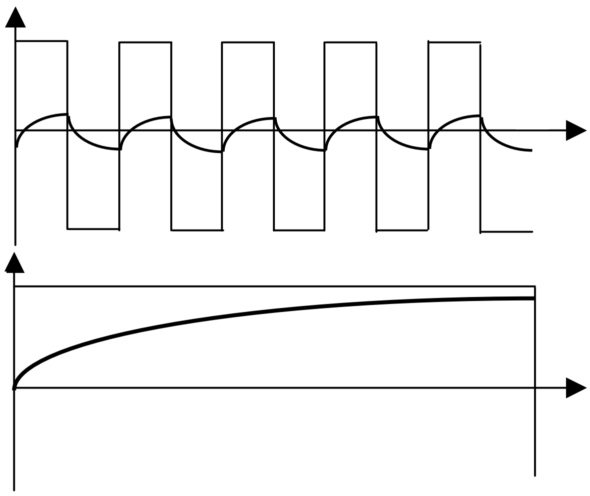
The voltage on the capacitance is

$$U_C(t) = U_n(1 - e^{-\frac{t}{T_{RC}}}), \tag{21}$$

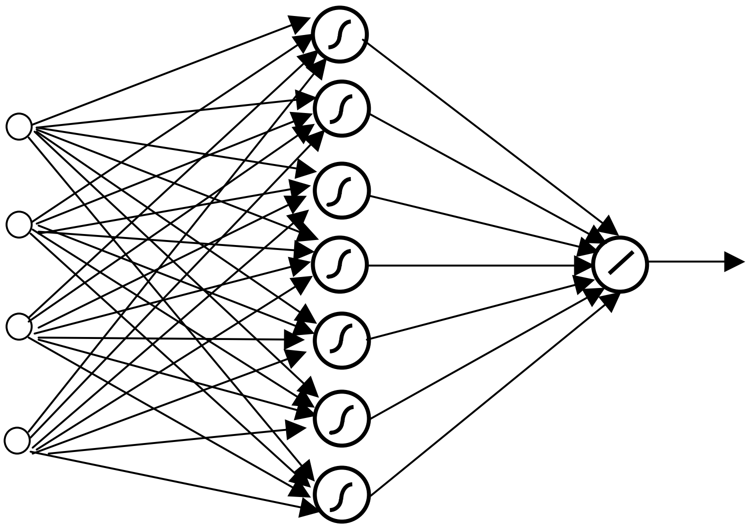
from where the RC time-constant is calculated as

$$T_{RC} = -\frac{t}{\ln\left[1 - \frac{U_C(t)}{U_n}\right]}. \tag{22}$$

Taking into account that $T_2$ is in the order of 100 µs – thus, $U_C(1ms) = 0.9U_n$ – and the variation of the steady-state voltage should be less than 10% of $U_n$ – that is, $U_C(50\mu s) = 0.1U_n$ – results that the time-constant of the RC circuit should be $T_{RC} = 0.5ms$. From equation (19) it can be seen that the rotor speed can be obtained by measuring the armature voltage, armature current and the derivative of the armature current. Because of the armature voltage variation, the armature current also has a fluctuation. Thus, before calculating the change of armature current we should also filter the armature current (we are not interested of the derivative of the fluctuation only in the trend of the armature current). Experiences show that better result can be achieved by the neural network when the derivative of the armature voltage is also supplied (in the same way as the armature current).



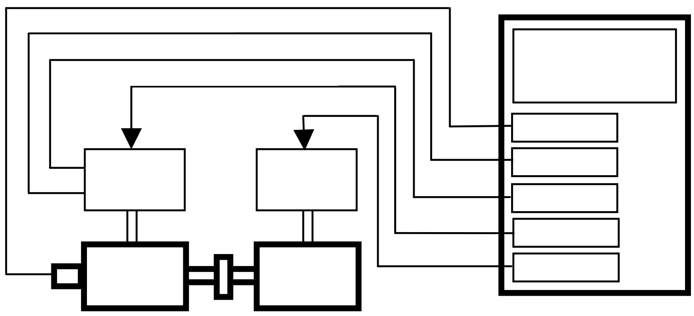
The time-constant of the RC circuit for the derivative of armature voltage and current is calculated on the bases of equation (21).



Thus, the inputs of the neural network are $u$, $\Delta u$, $i$, and $\Delta i$ while the output is the estimated rotor speed $\omega$. The neural network used is a multilayer feed-forward network with 5 hidden neurons (with tan-sigmoid transfer function) and 1 output neuron (with linear transfer function) as it can be seen in Figure 3. Using high-precision shaft encoder and PID controller, different rotor speed were set (negative and positive as well) and the obtained inputs ($u$, $\Delta u$, $i$, $\Delta i$) respective output ($\omega$) were registered. The multilayer network with the registered input/output training set was trained with the Levenberg-Marquardt back-propagation algorithm.

## 4 Hardware Realization

The experimental results are obtained using two identical permanent magnet DC motors which are joined together by a clutch as it is shown in Figure 4. One of them is used as a motor whose speed is controlled and the other is used as a load. Both of the DC motors are driven separately by two identical servo drivers containing analogue PI controllers for the current loop as well as a four-quadrant chopper amplifier working at 20 kHz. The neuro-fuzzy controller output is the current reference.



The neuro-fuzzy controller is implemented on a DSP (TMS320C31@50MHz) equipped with position encoder, 16 bit analogue-digital converters (ADC) and digital-analogue converters (DAC). The armature current is measured with the aid of a current transducer. The armature voltage is also measured with the aid of a current transducer using a resistor in parallel with the servomotor. The shaft encoder type is ROD426 and its precision is very low, $N_0$=2500 impulse/revolution. The impulses of the shaft encoder are processed by a PC extension card (EB 3005H).

The servomotor parameters are shown in Table 2.

Table 2
Parameters of the DC servomotor

| Parameters | Notation | Value | Unit |
|---|---|---|---|
| Nominal torque | $M_n$ | 3 | Nm |
| Nominal voltage | $U_n$ | 110 | V |
| Nominal current | $I_n$ | 13 | A |
| Maximal current | $I_{max}$ | 80 | A |
| Speed domain | $\omega$ | 0-2500 | rpm |
| Frictional torque | $M_f$ | 0.113 | Nm |
| Rotor inertia | $J_n$ | 0.00192 | $kgm^2$ |
| Torque coefficient | $K_n$ | 0.24 | Nm/A |
| Armature inductance | $L_a$ | 1.6 | mH |
| Armature resistance | $R_a$ | 0.49 | Ω |

Experimental results are presented in chapter 5.

# 5 The Neuro-fuzzy Controller

The armature current and the position is used by the instantaneous speed observer to estimate the speed at every $T_1 = 100\mu s$ . At every $T_2 = 10ms$ the position error is calculated and together with the output of the neural network corrects the estimated speed, and load.

Instead of a PID controller, as it is proposed by Hori, an adaptive fuzzy speed controller is used (see [5], [6] for more details) having two inputs, the error (difference between reference speed and estimated speed) and change of error, respectively. The output of the fuzzy controller is the current reference. The change of error is calculated as the difference of the momentary and previous error divided by $T_2$. The change of error should be calculated at $k$ =S before correcting the estimated speed in order to eliminate the appearance of a big change of error.

The fuzzy controller is adaptive in the sense that the system estimates the load with a very good precision and the error input to the fuzzy controller is given by:

$$E = \omega_a - \hat{\omega} + \alpha_3 \cdot \frac{T_2}{J_n} \hat{M}_d . \tag{23}$$

In this way the fuzzy controller becomes adaptive to the change of load, because when the direction of the load is such that the speed of the rotor decreases, the sign of the load in equation (23) is negative. Thus, the input error $E$ of the fuzzy controller will be increased, and the output of the fuzzy controller (reference signal) will be also increased. This results in a higher armature current which compensates the effect of the increased load. The same is true, when the direction of load is such that the rotor speed is increased. In this case the sign of the load in equation (23) will be positive. Thus, the input error $E$ of the fuzzy controller will be decreased, and the output of the fuzzy controller (reference signal) will be also decreased. This results in a lower armature current which compensates the effect of the load increasing the rotor speed.

Naturally, in this case the estimated rotor speed is corrected not only by the difference in position $\Delta\Theta$, but the output of the neural network, as well. Moreover, without neural network the difference in position $\Delta\Theta$ is divided almost evenly between the estimated speed and load, that is $\alpha_1 \cong \alpha_2$. In case of neural network, the difference in position $\Delta\Theta$ is mostly due to load estimation, thus, in this case $\alpha_1 \leq \alpha_2$.

In this way the load estimation is better, and in case of an abrupt load change, the controller does not need (3-4) $T_1$ times to estimate the changed load.

# 6 Results

Simulation result is shown in Figure 5, while experimental result can be seen in Figure 6.

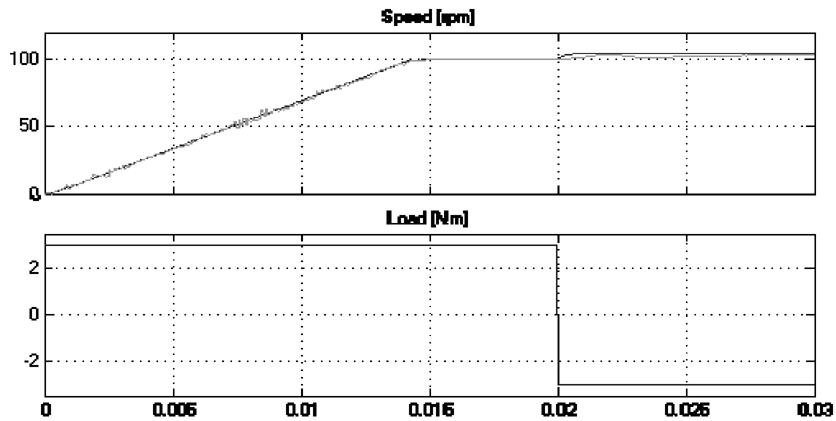



Figure 5

Step response of the simulation with MATLAB (load changes from +3 to -3 Nm at 0.02 ms)

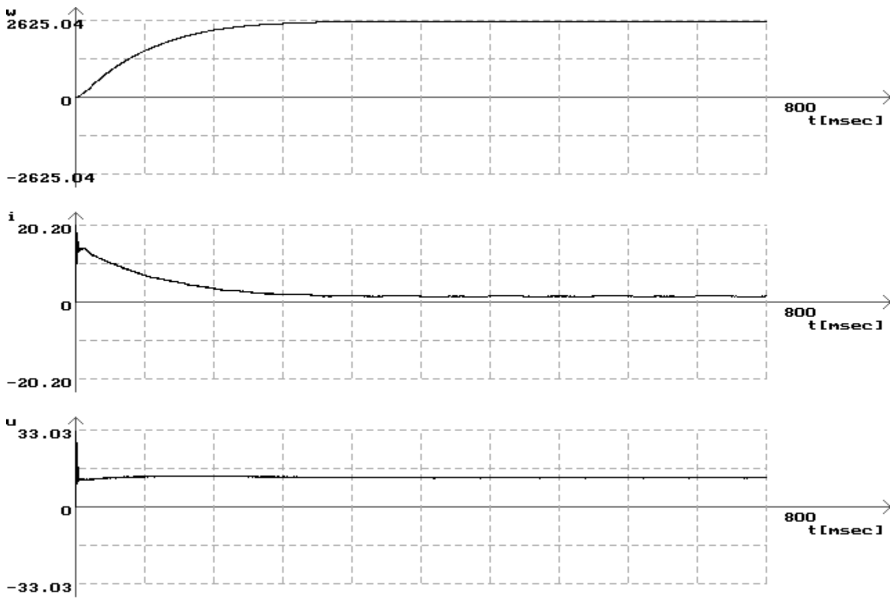



Figure 6

Step response of the experimental plant ($\omega_a$=300 rev/min)

Simulation was done using MATLAB and SIMULINK. As it can be observed in Figure 5 the load is changed abruptly at 0.02 ms from +3 Nm to -3 Nm. The estimated load follows the step change of the load within one $T_1$ period. Thus, the performance of the controller is increased.

Experimental results are in correlation with the simulation results and the technical considerations. When the speed estimation of the neural network is also used (see Figure 6), and the estimated load is corrected with a $\alpha_1 \leq \alpha_2$, the estimated load follows the change of load more quicker than at $\alpha_1 \cong \alpha_2$ when the estimation of speed and load is done only by using the difference position $\Delta\Theta$. The system under investigation showed good results even in lower ranges of reference speed.

## Conclusions

The PID controller proposed by Hori has the drawback that at abrupt change of load the controller needs at least (3-4)$T_1$ to estimate the load with a good precision. This is due to the fact that only one known value – difference in rotor position – is used to made corrections to two different estimated values. Thus, a second relation is needed to have a better estimation to both speed and load. This second relation uses the armature voltage and a feed forward neural network is used to model the proposed relation.

Simulation and experimental results have proven that the proposed controller has a better performance in estimating the change of load. The system under investigation shows good results even in lower ranges of reference speed.

**Acknowledgement**

This paper was supported by the Hungarian N.Sc. Fund (OTKA No. T 042866) for which the authors express their sincere gratitude.

**References**

[1] B. Kosko.: Neural Networks and Fuzzy Systems, Englewood Cliffs, NJ: Prentice Hall, Inc., 1992

[2] Y. Hori, et. al., An instantaneous speed observer for high performance control of DC servomotor using DSP and low precision shaft encoder, EPE'91, Vol. 3, Firenze, 1991, pp. 647-652

[3] Y. Hori: Robust and adaptive control of a servomotor using low precision shaft encoder, IEEE IECON'93, Hawaii, 1993, pp. 118-123

[4] F. Farkas, S. Halász: Adaptive Fuzzy Speed Controller for DC Drives Using Low Precision Shaft Encoder, EDPE'99, High Tatras, Slovakia, 1999, pp. 17-22

[5] F. Farkas, A. Zakharov, Sz. Varga: Speed and position controller for DC drives using fuzzy logic, Studies in Applied Electromagnetics and Mechanics (Vol. 16), Applied Electromagnetics and Computational technology II, Amsterdam, Netherlands, IOS Press, 2000, pp. 213-220

[6] F. Farkas, Sz. Varga, A. Zakharov: Investigation of DC servo drives with fuzzy logic control, Czasopismo Techniczne 4E/1998, Wydawnictwo Politechniki Krakowskiej, Poland, 1998, pp. 35-45